SOPHOS

SAV Interface Developer Toolkit, version 4.8
supplement

Document date: October 2008

Contents

1 About this supplement....3
2 SAVI configuration options....4
3 SAVI return values....21
4 SAVI storage IDs....28
5 Data types....32
6 Technical support....33
7 Copyright....34

1 About this supplement

This supplement documents the SAV Interface Developer Toolkit. It contains the following information:

- An overview and list of the SAVI (SAV Interface) configuration options.
- A list of SAVI return codes.
- A list of storage IDs.

The information in this supplement is valid for version 2.20 of the virus engine. Different options may become available for later versions. Check that you have the latest version of this document by visiting *www.sophos.com/support/docs*.

For full information about implementing SAVI, see the *SAV Interface Developer Toolkit user manual*.

To find out how to install SAVI on a Windows NT/2000/XP computer, see the *SAVI installation guide for Windows NT/2000/XP*.

2 SAVI configuration options

2.1 Introduction

This section contains a brief overview of SAVI configuration, along with a list of the current SAVI configuration options.

SAVI configuration options control

- the file types handled by SAVI
- the behaviour of SAVI as it scans files.

They are also used for other general configuration purposes.

Full information about how to use the member functions of the SAVI interfaces and enumerators is available in the *SAV Interface Developer Toolkit user manual*.

Note: The list of virus engine configuration options grows as more features and support for new file types is added to SAVI. Check that you have the latest version of this document by visiting *www.sophos.com/support/docs*.

2.2 Using SAVI configuration options

2.2.1 SAVI configuration options - general information

- Each SAVI configuration option has a unique **name** that is used to control the value of the associated SAVI feature (e.g. to turn on or off the handling of ZIP archives). SAVI configuration option names are case insensitive and don't contain spaces.

Names are passed to SAVI as an LPCOLESTR data type. Depending on the platform, this either maps to a pointer to a wide character string (16 bit, Unicode), or a char/multibyte string (8 bit). In the event of a mismatch in character encoding between SAVI and the client application, the client must carry out any necessary conversions.

- Each SAVI configuration option also has a unique **type**, one of the SOPHOS_TYPE_... codes listed in the header file savitype.h, under "Configuration option types". Currently-used configuration option types are listed in *SAVI configuration option data types* on page 5.

Values are passed in and out of SAVI as strings. This enables a number of different data types to be passed through a single API. Client applications must, where appropriate, convert numeric values to and from strings using appropriate conventions for the configuration option type. As with SAVI option names, the value string is an LPCOLESTR data type, and can therefore point to 8 or 16 bit data.

When changing a SAVI configuration option, both the name and the type must be passed through the SAVI API.

2.2.2 Local and Global Options

Most SAVI configuration options apply just to the SAVI object receiving the call to SetConfigValue. However, there is a set of options which are shared by all SAVI objects running in a process. If one of these options is changed then it will affect all SAVI objects, not just the one being called. These options all relate to aspects of SAVI which are also shared between the SAVI objects running in a process, in particular the threat data.

The global SAVI options are IdeDir, VirusDataDir, VirusDataName, AllowPartialVirusData and the entire set of ProductXxx options.

If, for example, the virus data directory is changed, this will flag that virus data needs to be reloaded from the new location. This reload will happen automatically on the next call to e.g. SweepFile, even if that is made on a different SAVI object to the one which called SetConfigValue.

A similar issue can occur if one of these options is changed followed a little later by a call to SetConfigDefaults. Depending on the timing of calls to SweepFile, this can result in multiple data reloads.

It should also be noted that, as a shared resource, care may be needed when changing these options on multiple SAVI object and/or threads. The majority of supported SAVI platforms have built-in thread synchronisation which will prevent conflicts here, but the designer of the client application needs to be aware of possible interactions between threads when using these options.

2.2.3 Listing and changing SAVI settings

- To ask a running SAVI object to list the names and types of configuration options it supports, use GetConfigEnumerator() to get a pointer to an enumerator interface. Then use the enumerator interface to obtain the list of configuration options supported.
- To configure engine settings to their default values, use the SetConfigDefaults() function of the ISavi2 or ISavi3 interfaces.
- To read and change the values of individual configuration options, use the SetConfigValue() and GetConfigValue() functions of the ISavi2 or ISavi3 interfaces.

For a full description of these functions, see the *SAV Interface Developer Toolkit user manual.*

2.3 SAVI configuration option data types

SAVI configuration option data types are listed under "Configuration option types" in the header file savitype.h, but not all of them are currently in use. The following is a list of currently-used data types.

SOPHOS_TYPE_U16	Unsigned 16-bit value. Passed to and from SAVI as a string representing the decimal value of the configuration option.

SOPHOS_TYPE_U32	Unsigned 32-bit value. Passed to and from SAVI as a string representing the decimal value of the configuration option. Many on/off settings are configured as SOPHOS_TYPE_U32, and are controlled by being set to 1 or 0 respectively.
SOPHOS_TYPE_OPTION_GROUP	A numeric value that may be 0, 1 or 2. Passed to and from SAVI as a string representing the value of the configuration option. See *SAVI group configuration options* on page 6 for more information.
SOPHOS_TYPE_STRING	This setting type represents a string, and so can be passed to and from SAVI directly, without needing any translation. **Note:** When calling SetConfigValue() for a setting of this type, SAVI takes a copy of the string, so the client may safely free the string buffer after the call.

2.4 SAVI group configuration options

2.4.1 Overview

Many configuration options have similar characteristics and can therefore be grouped, for example, into those that handle similar types of files. Group configuration options enable you to switch all the individual options in a group on or off with a single call to SetConfigValue().

This simplifies client code, which would otherwise have to make numerous calls to SetConfigValue(). It also means new configuration options that fall into a particular group are automatically controlled by the group configuration option, so you do not necessarily need to update SAVI client code to deal with new configuration options.

2.4.2 Currently defined groups

The following is a list of the currently defined group configuration options.

GrpSuper	Any option that is part of a group is also included in this group.
GrpArchiveUnpack	All archive and compressed archive file formats (e.g. ZIP, UUE, etc).
GrpSelfExtract	File formats that contain an executable stub that automatically decompresses the body of the file.
GrpExecutable	Executable files.

GrpInternet	File formats commonly in use on the internet.
GrpMSOffice	Office suite file formats from Microsfoft and other supported vendors.
GrpMisc	File formats that do not fall into any of the above categories.
GrpDisinfect	Enables or disables disinfection of all files for which disinfection is supported.
GrpClean	All "clean" file formats.
GrpWebArchive	Compression formats commonly used in HTTP and supported by web browsers.
GrpWebEncoding	HTML encoding schemes commonly used in web pages.

2.4.3 Using SAVI group configuration options

Group configuration options take one of three values:

- 0 represents **off**.
- 1 represents **on**.
- 2 (or greater) represents **invalid**.

invalid is not an error state. If passed into SAVI using SetConfigValue() it simply means none of the configuration options in the group will be altered.

SAVI also returns **invalid** if GetConfigValue() is called for one of the group configuration options (along with the informational return value SOPHOS_SAVI_INFO_OPT_GRP_INVAL_RTN).

Some individual options fall under the control of more than one group configuration option.

When configuring SAVI, first set the value of the group configuration option, then set the value of individual options. For example, if all archive formats except UUE are required, set GrpArchiveUnpack to **on**, then UueDecompression to **off**.

Note: Use of group configuration options is optional. SAVI can be configured just as well using individual options. Group options are available as a shortcut and to offer a degree of future-proofing in areas of SAVI configuration where precise control of every option is not critical.

2.5 Current SAVI configuration options

The following table lists the current SAVI configuration options.

Note: Not all settings are supported on all platforms.

The **Name** column lists both the name of the option and the version of the virus engine in which support for the option became available. Options that have been available since version 2.0 or earlier are listed as 'ver 2.0'.

The **Type** column lists the data type for each configuration option (listed in *SAVI configuration option data types* on page 5). Types are abbreviated as follows:

- U16 = (SOPHOS_TYPE_U16)
- U32 = (SOPHOS_TYPE_U32)
- GRP = (SOPHOS_TYPE_OPTION_GROUP)
- STR = (SOPHOS_TYPE_STRING)

The **Groups** column lists the groups, if any, the option belongs to (listed in *Currently defined groups* on page 6). Groups are abbreviated as follows:

- A = (GrpArchiveUnpack)
- C = (GrpClean)
- D = (GrpDisinfect)
- E = (GrpExecutable)
- I = (GrpInternet)
- M = (GrpMisc)
- N = (GrpWebEncoding)
- O = (GrpMSOffice)
- S = (GrpSelfExtract)
- W = (GrpWebArchive)

It can be assumed that any option that belongs to a group is also in GrpSuper.

The **Default** column lists the default value for the option.

Note: The EnableAutoStop option will abort the current scan if it seems to be requiring excessive system resources. On rare occasions it may be possible for an innocent file to trigger the AutoStop detection (a “false positive”). However, files causing SAVI to return ...SCAN_ABORTED should initially be treated in the same way as one containing a virus.

Name	Type	Description	Groups	Default
Access (ver 2.13)	U32	Enables scanning of databases saved in Microsoft Access (MDB) format.	O	1
ActiveMimeHandling (ver 2.2)	U32	Enables the detection of macro viruses within Microsoft Office documents saved in ActiveMime document format.	I	1

Name	Type	Description	Groups	Default
AllowPartialVirusData (ver 2.14)	U32	By default, errors in loading virus data are treated as fatal and scanning is not allowed. Setting this option to 1 means such errors are treated as warnings, and scanning using a partial virus data set is possible.		0
AppleSingle (ver 2.12)	U32	Enables scanning of Macintosh files saved as a single file in situations where support for Macintosh dual forks is not possible (e.g. MIME files).	A, I	1
ArjDecompression (ver 2.0)	U32	Enables decompression of Arj archives.	A	0
ASPack (ver 2.19)	U32	Enables scanning inside ASPack self-extracting executable archives.	S	1
Base64 (ver 2.18)	U32	Enables scanning of files encoded using the BASE64 scheme in situations where this is not flagged by, for example, a MIME header line.	M, I, N	0
BehaviourMalware (ver 2.38)	U32	Enable generic detection of malware based on Sophos Behavioural Genotype Detection. Note- detection of malware will be significantly reduced if this option is turned off.		1
BlockOnUpdate (ver 2.79)	U32	Controls whether the Sweep and Disinfection functions will block, or not, during an update. If non-blocking is selected, an error, SOPHOS_SAVI_ERROR_WOULDBLOCK is returned to the client. (W32 systems only)		1
Bzip2 (ver 2.17)	U32	Enables decompression of BZip2 archives.	A	0
CleanBmp, CleanGif, CleanJpeg, CleanPng, CleanRiff, CleanTiff (ver 2.17), CleanMp3 (ver 2.24), CleanMpeg (ver 2.25)	U32	Certain file formats (e.g. .bmp bitmap files) cannot contain viruses. These options enable files of the corresponding types to be positively identified and scanning of them to be stopped. This results in more efficient scanning of these file types.	C	1
ConcatenatedArchives (ver 2.27)	U32	Extends processing of multiple archive files which have been concatenated together so that they are all processed, even if they are		0

Name	Type	Description	Groups	Default
		separated by some junk data. NB can increase scan times slightly.		
CmzDecompression (ver 2.0)	U32	Enables decompression of Cmz archives.	A	0
DecompressVBA5 (ver 2.0)	U32	Decompresses the text part of a VBA5 project before scanning. This area is often corrupted, but can help positive identification of some viruses.	O	1
DelVBA5Project (ver 2.7)	U32	Causes the entire project containing any viral macros to be removed during disinfection.	-	1
Dmg (ver 2.49)	U32	Enables scanning of disk file images in Apple DMG format. Note - Scanning of DMG files also requires the appropriate file system plugin (e.g. Hfs+, see Hfs option) to be enabled.	-	-
DynamicDecompression (ver 2.0)	U32	Controls the ability to scan inside supported types of dynamically compressed executables. This type of file appears to be a normal executable. The main part of the program is compressed and extracted into memory at run-time. Sophos recommends this option should be turned on.	S	1
Elf (ver 2.10)	U32	Controls the module that interprets ELF binary files. This format is used for many Linux and FreeBSD executable files.	E	1
Emulation (ver 2.0)	U32	Enables the 16-bit x86 emulation engine, which assists in the detection of polymorphic executable file viruses. Sophos recommends this option should be turned on.	E	1
EnableAutoStop (ver 2.20)	U32	Causes scanning to be aborted for files (e.g. ""zip bombs"") whose characteristics cause the Virus Engine to consume excessive system resource (disc, memory, CPU). See note 1 at the end of this section.	-	0
Epoc (ver 2.45)	U32	Enables scanning of Symbian format mobile device executables.	E	1

Name	Type	Description	Groups	Default
ExcelFormulaHandling (ver 2.0)	U32	Enables scanning of Excel formulas for known formula viruses. Formulas occupy cells on the worksheet and should not be confused with macros.	O	1
ExecFileDisinfection (ver 2.4)	U32	Some executable file viruses can be removed from files fairly safely. This option enables that process. Sophos recommends that infected executables are replaced from the original installation disks as soon as possible.	D	1
ExtensiveScan (ver 2.48)	U32	Normal scanning behaviour is to only switch to ""extensive"" scan mode when initial file analysis indicates that this is necessary (e.g. strangely formed EXE header). This option overrides the tests and performs a full ""extensive"" scan on every file. Note - Enabling this option will result in scans taking longer and could generate an occasional false positive report of a threat. Sophos recommends that this option is only enabled when performing an in-depth analysis of suspicious files.	-	0
Fsg (ver 2.19)	U32	Enables scanning inside FSG self-extracting executable archives.	S	1
FullMacroSweep (ver 2.0)	U32	Enables a fallback mechanism that has been deployed in the past when new file structures are encountered. Sophos recommends that you keep this option turned off.	-	0
FullPdf (ver 2.17)	U32	Enables every part of a PDF file to be scanned, not just the embedded file streams. This takes longer.	-	0
FullSweep (ver 2.0)	U32	Enables full sweeping, which performs extensive fixed pattern matching as well as the standard intelligent virus search. Turning on this option may have severe performance implications.	-	0
GzipDecompression (ver 2.17)	U32	Enables decompression of Gzipped archives.	M, W	0

Name	Type	Description	Groups	Default
HelpHandling (ver 2.0)	U32	Enables scanning of files embedded in Microsoft Help files.	M	1
Hfs (ver 2.49)	U32	Enables support for the Apple HFS+ file system as often encountered in DMG files (see also Dmg option).	-	0
Http (ver 2.50)	U32	Enables the scanning of HTTP streams	I	0
HqxDecompression (ver 2.0)	U32	Controls the ability to extract files from BinHex archives.	A	0
Html (ver 2.0)	U32	Controls the module that optimises scanning in HTML files.	I	1
IdeDir (ver 2.14)	STR	The passed string defines the full path name of the directory in which SAVI should search for IDE files. The name must not have a trailing separator character.	-	
IgnoreTemplateBit (ver 2.0)	U32	Microsoft Office documents that contain executable macros will usually have a single bit in the file header set to 1 to indicate the presence of executable code. Under normal circumstances macros contained in a file that does not have this bit set are not able to execute. If this option is turned off, the Virus Engine will scan only files with this template bit set to 1. Sophos recommends that you keep this option turned on.	-	1
ISCabinet (ver 2.17)	U32	Enables decompression of Cabinet files created using the InstallShield compression format (e.g. _sys1.cab).	A	0
ISCabinetFull (ver 2.17)	U32	Increases detection reliability in old format InstallShield files, but with a slight performance overhead.	-	0
ITSS (ver 2.0)	U32	Enables scanning of Microsoft Compressed Help files (.chm, .hxs).	A, W	0
Java (ver 2.13)	U32	Enables scanning of Java class files.	E, I	1
Lha (ver 2.2)	U32	Enables decompression of Lha archives.	A	0

Name	Type	Description	Groups	Default
LoopBackEnabled (ver 2.0)	U32	Controls SAVI's ability to scan inside certain types of container file (e.g. FAT container files or disk image files such as are found in CD boot sectors or on OpenVMS).	M	0
MachO (ver 2.17)	U32	Enables scanning of Mach-O executables (used on Mac OS X). Note that "O" is the letter, not a zero.	E	1
MaxRecursionDepth (ver 2.0)	U16	Controls the maximum number of times the engine will recurse into archives when they are found (e.g. when a zip file is compressed within another zip file). The maximum value is 100, but SAVI internally has some extra levels of recursion to cover internal data streams found within some file types.	-	16
MaxIntRecDepth (ver 2.24)	U16	As per MaxRecursionDepth but applying to internal data streams.	-	25
MbinDecompression (ver 2.0)	U32	Controls the ability to look inside MacBinary archives.	A	0
Mbox (ver 2.42)	U32	Enables scanning of Unix 'mbox' format mail stores.	I	0
Mime (ver 2.10)	U32	Enables decoding and scanning inside a MIME encoded message block.	I, N	0
MimeEmbedded (ver 2.29)	U32	To enable scanning of email embedded inside another email rather than an attachment.	-	1
MimeEmbedLimit (ver 2.31)	U16	Maximum number of embedded file streams that can be extracted from a MIME file.	-	25
MimeEmbedLines (ver 2.32)	U16	The maximum number of lines MIME source to scan looking for embedded objects (inline BASE64).	-	500
MimeRescan (ver 2.29)	U32	Enable the Virus Engine to rescan a MIME encoded object as a single object after scanning the individual message parts. The default value of 2 means 'automatic' i.e. the Virus Engine will decide when to temporarily enable the option while scanning MIME.	-	2

Name	Type	Description	Groups	Default
MSCabinet (ver 2.0)	U32	Enables decompression of Microsoft Cabinet files (cab). (Partial files at the beginning and end of multi-part cabinet files will not be fully scanned).	A	0
MSCompress (ver 2.2)	U32	Controls the ability to scan inside files compressed with the Microsoft compression utility.	A	0
Msi (ver 2.29)	U32	Enable scanning of MS Installer files. This needs other options, such as MSCabinet, to be set to be effective depending on the file content	-	0
NamespaceSupport (ver 2.0)	U32	Controls handling of Macintosh (resource + data fork) files and archives (on platforms where these are supported). Settings are SOPHOS_MAC_FILES (enable scanning for Mac viruses), SOPHOS_DOS_AND_MAC_FILES (also scan data fork) or SOPHOS_DOS_FILES (disable Mac file scanning). Note - (1) default value is platform dependent. (2) Enabling data fork scanning carries a slight performance overhead and so is only recommended on platforms where OS8/OS9 viruses may be encountered.	-	
Odoc (ver 2.40)	U32	Enables support for ""Open Document"" format, (generally consisting of XML within a zip archive). This includes both StarOffice/OpenOffice and Microsoft OpenXML formats (see Sdoc and Oxml options).	O	1
Office2001Handling (ver 2.10)	U32	Controls the ability of the engine to understand Microsoft Office 2001 files. This version of Office was only released on the Macintosh platform.	O	1
OF95DecryptHandling (ver 2.0)	U32	Password-protected Office 95 documents are encrypted with a very simple algorithm. Enabling this option causes the virus engine to break this encryption to search for viruses.	O	1

Name	Type	Description	Groups	Default
OleDataMsoHandling (ver 2.13)	U32	Enables scanning of files saved in HTML format from Microsoft Office.	O	1
OleRawHandling (ver 2.17)	U32	Enables scanning of other OLE types (e.g. messages saved in Microsoft Outlook).	I, O	1
OleScriptHandling (ver 2.13)	U32	Enables scanning of Access macros.	O	1
Ole2FileDisinfection (ver 2.10)	U32	This option enables disinfection of OLE2 (Microsoft Office) files to be configured independently of other types of file disinfection.	D	1
Ole2Handling (ver 2.0)	U32	Enables the intelligent scanning of OLE2 (Microsoft Office) format documents. Sophos recommends that this option should be turned on.	O	1
OpenMacRf (ver 2.17)	U32	Enables detection and opening of Mac resource forks in all file types scanned.	-	1
OutlookExpress (ver 2.10)	U32	Enables scanning of files contained within Outlook Express mailboxes. NB: to scan email attachments, also set the Mime option.	I	0
Oxml (ver 2.44)	U32	Enable scanning of Microsoft Open XML documents (use in conjunction with Odoc option).	O	1
PalmPilotHandling (ver 2.2)	U32	Enables the scanning of Palm Pilot format files (.prc).	M	1
Pdf (ver 2.10)	U32	Controls the PDF file interpretation module	M	1
PECompact (ver 2.19)	U32	Enables scanning inside PECompact self-extracting executable archives.	S	1
PeEmulator (ver 2.13)	U32	Enables the 32 bit executable emulator. This is required to support some advanced virus detection modes in 32 bit executable files.	E	1
PEHandling (ver 2.0)	U32	Enables the intelligent scanning of Win32/PE format files. This is the standard format for Windows executable binaries such as .exe and .dll.	E	1

Name	Type	Description	Groups	Default
PowerPointEmbedded Handling (ver 2.0)	U32	Enables the scanning of files embedded within Microsoft PowerPoint presentations.	O	1
PowerPointMacro Handling (ver 2.0)	U32	Enables the scanning of macros in PowerPoint presentations.	O	1
Product CLI, ProductDesktop, ProductGateway, ProductMobile, ProductUnspecified, ProductWeb (ver 2.47)	U32	These options enable identities targetted at specific product types. ProductCli - identities appropriate to a command-line scanner. ProductDesktop - identities appropriate to a desktop product. ProductGateway - identities appropriate to a gateway scanner. ProductMobile - identities for threats which only affect mobile products e.g. smart phones. ProductUnspecified - non-specific identities. This includes all traditional threats: viruses, worms, trojans, PUAs etc. ProductWeb - identities specific to web gateway products. Note- Sophos strongly recommends that ProductUnspecified is not set to 0, as this will effectively disable most of the threat detection offered by SAVI. Product-specific options are targetted at the relevant products and inappropriate use may impact on scanning performance.	-	0, 0, 0, 1, 1, 0
ProjectHandling (ver 2.0)	U32	Enables the scanning of VBA macros in Microsoft Project.	O	1
RarDecompression (ver 2.0)	U32	Enables decompression of RAR archives.	A	0
Rpm (ver 2.17)	U32	Enables the scanning of distribution archives in RedHat Package Manager format. NB: to scan .rpm files, set the GZipDecompression and UnixArchive options as well.	A	0
Rtf (ver 2.10)	U32	Enables scanning for objects that may be embedded inside Rich Text Format files.	O, M	1
Saveset (ver 2.30)	U32	Enable scanning of VMS 'saveset' archives	-	0
ScrapObjectHandling (ver 2.0)	U32	Enables the handling of scrap objects that are used in the transfer of data between certain applications.	O	1

Name	Type	Description	Groups	Default
Sdoc (ver 2.44)	U32	Enable scanning of StarOffice /OpenOffice documents (use in conjunction with Odoc option).	O	1
SfxArchives (ver 2.0)	U32	Enables the virus engine to search in supported types of self-extracting archive.	A, S	0
Sis(ver 2.44)	U32	Enable scanning of 'SIS' (Symbian Installation System) files as used on some mobile devices.	A	0
Skip (ver 2.18)	U32	Used for Virus Engine internal control. Should be left on (value 1) by SAVI applications.	-	1
SrpStreamHandling (ver 2.0)	U32	Deals with another type of data within Office documents.	O	1
StorageDetOnly (ver 2.0)	U32	Virus detection has two stages. First a file is identified and classified according to its type. In the case of archive files, the individual files within the archive are extracted and treated in the same way. Then scanning is then carried out according to the file type. When StorageDetOnly is enabled, virus scanning is not carried out. The virus engine simply identifies the file types and returns information about any files contained inside.	-	0
StorageReport (ver 2.0)	U32	When enabled, zero, one or more calls are made to the OnClassification() callback function in INotify2 for each filename passed to OnFileFound().	-	0
StorageReportAll (ver 2.17)	U32	By default, only the published storage types are notified via the OnClassification() callback. This option enables intermediate and weakly classified storage types to be reported too. Many of these are internal to the Engine and will not be documented.	-	0
Stuffit (ver 2.17)	U32	Enables detection and reporting of some types of Stuffit archives.	-	0
StrictPdf (ver 2.36)	U32	Causes SAVI to report ...FORMAT_NOT_SUPPORTED if the Virus Engine encounters data in a PDF file encoded	-	0

Name	Type	Description	Groups	Default
		with an unrecognised 'custom filter'. Default is to scan the stream without decoding it.		
StrongPdf (ver 2.36)	U32	Causes SAVI to report... CORRUPT if it encounters sections within a PDF file which cannot be interpreted by the Virus Engine.	-	0
Szip (ver 2.50)	U32	Enable Seven Zip support.	A	0
TarDecompression (ver 2.0)	U32	Enables scanning inside Unix tar archives.	A	0
TnefAttachment Handling (ver 2.18)	U32	Enables decoding of files encoded with the Microsoft TNEF format, used by some mail client applications.	I	0
TnefEmbedHandling (ver 2.18)	U32	Enables scanning of TNEF files embedded within another mail file. The Microsoft TNEF format is used by some mail client applications.	-	0
TrueFileTypeDetection (ver 2.70)	U32	Reports the real type of the file		0
TrueFileTypeDetectionLevel (ver 2.70)	U16	Set file type reporting mode. When more than one file type is detectable when examining a file, SAVI reports just one or more file types depending on the TFT level set. SOPHOS_TFT_LEVEL_ONEFILETYPE Having found and reported one, the most accurate file type, SAVI will not check for further file types. SOPHOS_TFT_LEVEL_ALLFILETYPES Having found and reported one file type, SAVI will continue to check for further file types with the same and lower priority. SOPHOS_TFT_LEVEL_ALLHIGHESTPRIORITY Having found and reported one file type, SAVI will continue to check for further file types with the same priority. The check for lower priority file types will not be performed. This will reduce the number of reports that would otherwise be reported using		1

Name	Type	Description	Groups	Default
		SOPHOS_TFT_LEVEL_ALLFILETYPES option.		
UnixArchive (ver 2.17)	U32	Enables scanning inside Unix archive types (ar, cpio).	A	0
Upx (ver 2.7)	U32	Enables scanning inside UPX-compressed self-extracting executable archives.	S	1
UTF16 (ver 2.17)	U32	Enables scanning of web pages created using UTF16 (Unicode) character encoding.	I	1
UueDecompression (ver 2.0)	U32	Enables the virus engine to decode UUEncoded files. The UUEncode portion must start within the first 4k of the document.	A	0
VBA3Handling (ver 2.0)	U32	Enable the scanning of versions of Microsoft Visual Basic for Applications embedded in supported file types, like Office and Visio documents.	O	1
VBA5Handling (ver 2.0)	U32	Enable the scanning of versions of Microsoft Visual Basic for Applications embedded in supported file types, like Office and Visio documents.	O	1
Vba5Dir (ver 2.22)	U32	Option currently unimplemented	O	0
VbaOnly (ver2.25)	U32	Enable scanning of files containing only Visual Basic for Applications (e.g. PowerPoint PPA files).	O	0
VbaTable (ver 2.32)	U32	Enables scanning of Word document table streams.	-	1
Vba5p (ver 2.17)	U32	Enables detection of macros in p-code (compiled Visual Basic) as well as in Visual Basic source code.	O	1
Vbe (ver 2.10)	U32	Enables the scanning of encoded VBScript files.	I	1
VbFiltering (ver 2.13)	U32	Enables canonicalisation of Visual Basic code. This increases the detection efficiency of the other Visual Basic for Applications (VBA) plugins.	I	0

Name	Type	Description	Groups	Default
VirusDataDir (ver 2.0)	STR	The passed string defines the full path name of the directory in which the main virus data file is located. The name should not have a trailing directory separator character.	-	-
VirusDataName (ver 2.0)	STR	The passed string defines the name of the virus data file. This name should not include a directory component (this is defined by the VirusDataDir setting) or a file extension. For example, if the main virus data file name is vdl.dat then this string should be passed as vdl.	-	-
VisioEmbedHandling (ver 2.20)	U32	Enable scanning of files embedded inside Microsoft Visio documents	0	0
VisioFileHandling (ver 2.4)	U32	Enables scanning of macros inside Microsoft Visio files.	O	1
WordB (ver 2.0)	U32	Enables scanning of WordBasic.	O	1
Xml (ver 2.18)	U32	Enables scanning of XML files. Note - scanning of files saved in XML format by Microsoft Office 11+ is enabled using the XmlOdoc option.	-	0
XmlOdoc (ver 2.44)	U32	Enables detection of Open Document files (and subsequent processing) without requiring the XML plugin (Xml option) to be enabled first.	O	1
ZipDecompression (ver 2.0)	U32	Enables scanning inside Zip archives.	A, W	0
ZipUseChd (ver 2.17)	U32	Improves reliability of scanning of Zip files created in a non-standard way (local file directory invalid).	-	1

3 SAVI return values

All SAVI interface functions return values that indicate whether the function succeeded.

This table lists the least significant word of the return value. The complete numerical values corresponding to these symbols may vary from one platform to another. Numerical values are defined in the file savierr.h.

SAVI client applications should be designed to behave gracefully if they encounter new return codes not listed in the header file used at the time the application was compiled.

The current full list of return values is as follows.

Value	Code	Explanation
0000	SOPHOS_S_OK	The function call succeeded.
0001	SOPHOS_S_FALSE	The function call partly succeeded. The exact meaning depends on the function. Refer to the function details in the section on the relevant SAVI interface.
000E	SOPHOS_E_OUTOFMEMORY	The function could not complete successfully because it ran out of memory.
0057	SOPHOS_E_INVALIDARG	The value of an argument supplied to the function by the SAVI client is invalid.
0070	SOPHOS_E_OUT_OF_DISK	SAVI encountered a problem while trying to create or write to a temporary file. Check permissions/available disk space in the SAVI virus engine temporary directory.
010E	RPC_E_WRONG_THREAD	A callback was installed using RegisterNotification() on a different thread to the

Value	Code	Explanation
		current one. Both calls must be on the same thread.
0110	SOPHOS_CLASS_E_NOAGGREGATION	If CreateInstance() (i.e. the first parameter to IClassFactory()) is not NULL, this value is returned.
4001	SOPHOS_E_NOTIMPL	The function was not implemented.
4002	SOPHOS_E_NOINTERFACE	The caller specified an interface type (REFIID) which is unknown or not supported.
FFFF	SOPHOS_E_UNEXPECTED	An unexpected error occurred.
0200	SOPHOS_SAVI_ERROR_INITIALISING	The SAVI interface could not be initialised.
0201	SOPHOS_SAVI_ERROR_TERMINATING	The SAVI interface could not be terminated.
0202	SOPHOS_SAVI_ERROR_SWEEPFAILURE	An error occurred during a virus scan.
0203	SOPHOS_SAVI_ERROR_VIRUSPRESENT	A virus was found during a virus scan.
0204	SOPHOS_SAVI_ERROR_NOT_INITIALISED	A function was called when the interface was not initialised.
0205	SOPHOS_SAVI_ERROR_IC_INCOMPATIBLE_VERSION	The version of the on-access scanner (InterCheck) installed is not compatible with SAVI.
0206	SOPHOS_SAVI_ERROR_IC_ACCESS_DENIED	The SAVI client process has insufficient rights to disable on-access scanning (InterCheck). The scan could not be performed.

Value	Code	Explanation
0207	SOPHOS_SAVI_ERROR_IC_SCAN_PREVENTED	The on-access scanning (InterCheck) client is active and cannot be disabled. The scan could not be performed.
0208	SOPHOS_SAVI_ERROR_DISINFECTION_FAILED	An attempt to disinfect a file or sector failed.
0209	SOPHOS_SAVI_ERROR_DISINFECTION_UNAVAILABLE	An attempt to disinfect a file or sector failed because disinfection was not available.
020A	SOPHOS_SAVI_ERROR_UPGRADE_FAILED	The setup program was executed to upgrade the Sophos Anti-Virus installation, but it did not complete successfully. SAVI will not function until setup is run successfully.
020B	SOPHOS_SAVI_ERROR_SAV_NOT_INSTALLED	SAVI could not locate a version of Sophos Anti-Virus to use.
020C	SOPHOS_SAVI_ERROR_INVALID_CONFIG_NAME	An invalid configuration setting name was supplied.
020D	SOPHOS_SAVI_ERROR_INVALID_CONFIG_TYPE	An invalid configuration setting type was supplied.
020E	SOPHOS_SAVI_ERROR_INIT_CONFIGURATION	An internal error occurred during an attempt to access SAVI configuration information.
020F	SOPHOS_SAVI_ERROR_NOT_SUPPORTED	SAVI has encountered an unrecognised form of one of the file types it supports and so is unable to scan it. This error code is also returned when a call is made to a function which

Value	Code	Explanation
		is not supported by this implementation of SAVI.
0210	SOPHOS_SAVI_ERROR_COULD_NOT_OPEN	An error occurred during an attempt to access the item passed to SAVI for scanning.
0211	SOPHOS_SAVI_ERROR_FILE_COMPRESSED	The file was compressed but no virus was found at the outer level.
0212	SOPHOS_SAVI_ERROR_FILE_ENCRYPTED	The file was encrypted.
0213	SOPHOS_SAVI_ERROR_INFORMATION_NOT_AVAILABLE	The additional virus location is unavailable.
0214	SOPHOS_SAVI_ERROR_ALREADY_INIT	An attempt was made to initialise the SAVI interface when it was already initialised.
0215	SOPHOS_SAVI_ERROR_STUB	There was an attempt to use a stub library.
0216	SOPHOS_SAVI_ERROR_BUFFER_TOO_SMALL	The caller supplied a character buffer that was too small to contain the information requested.
021A	SOPHOS_SAVI_ERROR_CORRUPT	A scan could not proceed because a file or sub-file was corrupted.
021B	SOPHOS_SAVI_ERROR_REENTRANCY	An attempt was made to re-enter the SAVI interface from within a notification callback. This is not permitted.
021C	SOPHOS_SAVI_ERROR_CALLBACK	An error occurred using a notification callback within a SAVI client.
021D	SOPHOS_SAVI_ERROR_PARTIAL_INFORMATION	GetVirusEngineVersion() succeeded, but not all of

Value	Code	Explanation
		the requested data was available.
021E	SOPHOS_SAVI_ERROR_OLD_VIRUS_DATA	A call to initialise the SAVI DLL succeeded but the internal virus data was old. Update virus data as soon as possible (by updating Sophos Anti-Virus).
021F	SOPHOS_SAVI_ERROR_INVALID_TMP	No valid temp directory was found.
0220	SOPHOS_SAVI_ERROR_MISSING_MAIN_VIRUS_DATA	The main body of virus data is missing.
0221	SOPHOS_SAVI_INFO_IC_ACTIVE	The InterCheck client is active and could not be disabled. This may cause calls to some Sweep...() and Disinfect...() functions to fail.
0222	SOPHOS_SAVI_ERROR_VIRUS_DATA_INVALID_VER	The virus data main body has an invalid version number.
0223	SOPHOS_SAVI_ERROR_MUST_REINIT	SAVI must be reinitialised because the virus engine is a later version than the version of SAVI currently running.
0224	SOPHOS_SAVI_ERROR_CANNOT_SET_OPTION	Cannot set option value as the value of the virus engine setting is immutable.
0225	SOPHOS_SAVI_ERROR_PART_VOL	The file passed for scanning represented part of a multi-volume archive. The file cannot be scanned completely.
0227	SOPHOS_SAVI_INFO_OPT_GRP_INVAL_RTN	Returned if GetConfigValue() is called

Value	Code	Explanation
		for a grouped engine setting. This indicates that no meaning can be assigned to the value returned for the setting (see section 1.4.3).
0228	SOPHOS_SAVI_ERROR_VDLD_ACTIVITY	The operation failed due to an incompatible pending or ongoing activity on virus data (e.g. an attempt to scan a file while updating VDL data, or an attempt to update VDL data while scan is in progress).
0229	SOPHOS_SAVI_ERROR_STREAM_READ_FAIL	For ISaviStream implementation: ReadStream failed.
022A	SOPHOS_SAVI_ERROR_STREAM_WRITE_FAIL	For ISaviStream implementation: WriteStream failed.
022B	SOPHOS_SAVI_ERROR_STREAM_SEEK_FAIL	For ISaviStream implementation: SeekStream failed.
022C	SOPHOS_SAVI_ERROR_STREAM_GETLENGTH_FAIL	For ISaviStream implementation: GetLength failed.
022D	SOPHOS_SAVI_ERROR_MISSING_VDL_PART	One of the files or sections in a split virus data set could not be located or an error was encountered while reading it.
022E	SOPHOS_SAVI_WARNING_MISSING_VDL_PART	Warning only version of the previous code.
022F	SOPHOS_SAVI_ERROR_VDL_CHECKSUM	One of the files in a split-virus data set has the wrong checksum.

Value	Code	Explanation
0230	SOPHOS_SAVI_WARNING_VDL_CHECKSUM	Warning only version of the previous code.
0231	SOPHOS_SAVI_ERROR_SCAN_ABORTED	Scan aborted (see ""EnableAutoStop"" option, section 1).
0232	SOPHOS_SAVI_WARNING_INFO_UNDEFINED	The item of information requested is not defined for this object.
0236	SOPHOS_SAVI_ERROR_INTERNAL_ERROR	A serious internal error has occurred. The SAVI object should be released and a new one created and initialised.
0237	SOPHOS_SAVI_ERROR_RECURSION_LIMIT	The scan has been terminated due to the Virus Engine reaching its storage recursion limit (e.g.files nested inside other files).
0203	SOPHOS_SAVI_INFO_THREATPRESENT	This return code is just an alias for ... VIRUSPRESENT but uses a more generic term to reflect the wider range of threats now detected by SAVI.

4 SAVI storage IDs

The ISweepNotify2 interface includes an OnClassification() function, which can return information about the types of file and data formats identified by the virus engine as it scans. For further details of ISweepNotify2, refer to the *SAV Interface Developer Toolkit user manual*.

A "storage ID" is assigned to each kind of file or data format identified. These IDs are defined in the header file savitype.h, and are listed below. The list is updated as the virus engine is developed to analyse more file types. Applications should be designed to cope with being passed storage IDs which aren't on the list below.

Also listed below are some IDs that are not defined in savitype.h and are not normally reported, either because they are used only internally by SAVI or because classification by the virus engine is weak (i.e. less than 100% confidence). To enable reporting of these IDs too, use the StorageReportAll SAVI configuration option described in *Current SAVI configuration options* on page 7.

Storage ID name	Value (hex)	Description
ID_OLE2_STORAGE	20	OLE2 file
ID_SARC_ZIP_STORAGE	30	Zip archive
ID_SARC_TAR_STORAGE	31	TAR archive
ID_SARC_GZIP_STORAGE	32	GZip archive
ID_SARC_ARJ_STORAGE	33	ARJ archive
ID_SARC_RAR_STORAGE	34	RAR archive
ID_SARC_UUE_STORAGE	35	UUE archive (not normally reported)
ID_SARC_CMZ_STORAGE	36	CMZ archive
ID_SARC_PP97_STORAGE	37	Compressed PowerPoint 97 (not normally reported)
ID_SARC_HQX_STORAGE	38	Mac Binhex
ID_SARC_MBIN_STORAGE	39	MacBinary file
ID_SARC_CAB_STORAGE	3a	Microsoft Cabinet archive
ID_SARC_TNEF_STORAGE	3b	MS Mail winmail.dat file
ID_SARC_LHA_STORAGE	3c	LHA archive
ID_SARC_MS_STORAGE	3d	MsCompress file

Storage ID name	Value (hex)	Description
ID_SARC_MSO_STORAGE	3e	MSO/Active MIME
ID_SARC_APPLE_STORAGE	3f	AppleSingle/AppleDouble packaged file
ID_SARC_PDF_STORAGE	41	PDF (not normally reported)
ID_SARC_BZIP2_STORAGE	42	BZip2 archive
ID_SARC_STF5_STORAGE	43	Stuffit version 7 (compression version 5) archive
ID_SARC_STF1_STORAGE	44	Older format Stuffit archive
ID_SARC_ICAB_STORAGE	45	InstallShield Cabinet archive
ID_SARC_ITSS_STORAGE	46	Microsoft Compressed Help file
ID_SARC_STF8_STORAGE	47	Stuffit version 8 archive
ID_ODOC_STORAGE	48	XML-based Office file format
ID_SARC_SIS_STORAGE	49	SIS archive
ID_SARC_SPARSETAR_STG	4c	'Sparse' tar file
ID_SEXP_DIET_STORAGE	50	DIET self-extracting executable
ID_SEXP_PKLT_STORAGE	51	PKLite self-extracting executable
ID_SEXP_LZEX_STORAGE	52	LZEX self-extracting executable
ID_SEXP_UPX_STORAGE	53	UPX self-extracting executable
ID_SEXP_PETITE_STORAGE	54	Petite self-extracting executable
ID_SEXP_ASPACK_STORAGE	55	ASPack self-extracting executable
ID_SEXP_FSG_STORAGE	56	FSG self-extracting executable
ID_SEXP_PEC_STORAGE	57	PECompact self-extracting executable
ID_SFX_STORAGE	58	Self-extracting archive
ID_CONCAT_STORAGE	59	Concatenated archive file
ID_EXEC_STORAGE	60	DOS/Windows executable file
ID_ELF_STORAGE	68	Unix/Linux executable file
ID_MACHO_STORAGE	6d	Mach-O executable file

Storage ID name	Value (hex)	Description
ID_EPOC_STORAGE	7c	EPOC executable
ID_HELP_STORAGE	90	Windows Help file
ID_CLEAN_JPG_STORAGE	91	JPG image file
ID_CLEAN_BMP_STORAGE	92	Bitmap image file
ID_CLEAN_GIF_STORAGE	93	GIF image file
ID_CLEAN_RIFF_STORAGE	94	RIFF media file
ID_CLEAN_TIFF_STORAGE	95	TIFF image file
ID_CLEAN_PNG_STORAGE	96	PNG image file
ID_MP3_STORAGE	97	MP3 audio file
ID_MPEG_STORAGE	98	MPEG video file
ID_LPBK_STORAGE	a0	“Loopback” (encoded file system) file
ID_COMP_WORD_STORAGE	b0	Word Basic macros
ID_COMP_VBA3_STORAGE	b1	Excel 95 macros
ID_COMP_VBA5_STORAGE	b2	Visual Basic as used in Office 97 or later
ID_COMP_VB5D_STORAGE	b3	Processed ID_COMP_VBA5_STORAGE type (SAVI internal type)
ID_COMP_XF95_STORAGE	b4	Excel 95 formulae
ID_COMP_XF97_STORAGE	b5	Excel 97 formulae
ID_COMP_PP97_STORAGE	b6	PowerPoint 97 (not normally reported)
ID_COMP_SCRP_STORAGE	b8	Embedded document in OLE2 file (not normally reported)
ID_COMP_VISIO_STORAGE	b9	Visio file (not normally reported)
ID_COMP_VB5P_STORAGE	ba	Visual Basic p-code
ID_MIME_STORAGE	d0	MIME encoding
ID_BASE64_STORAGE	d1	Base64 encoding
ID_RTF_STORAGE	d4	RTF file

Storage ID name	Value (hex)	Description
ID_VBE_STORAGE	d8	Encoded Visual Basic file
ID_HTML_STORAGE	d9	HTML file
ID_OEDBX_STORAGE	da	Outlook Express (Windows) file
ID_OEMAC_STORAGE	db	Outlook Express (Mac) file
ID_UTF16BE_STORAGE	dd	Big-endian UTF16 character encoding
ID_UTF16LE_STORAGE	de	Little-endian UTF16 character encoding
ID_MBOX_STORAGE	df	Unix 'mbox' mail store file
ID_MAC_STORAGE	f0	Mac data fork (not normally reported)
ID_MAC_RES_STORAGE	f3	Mac resource fork (not normally reported)
ID_PRC_RES_STORAGE	f4	Palm OS resource file (not normally reported)
ID_JAVA_STORAGE	f5	Java byte code class file
ID_ACCESS_STORAGE	f6	Access database file (MDB format)
ID_UNIXARCHIVE_STORAGE	f7	Unix ar or cpio archive
ID_RPM_STORAGE	f8	RedHat Package Manager file
ID_XML_STORAGE	f9	XML file
ID_XMLODOC_STORAGE	fa	OpenDocument file (StarOffice /OpenOffice)
ID_HFS_STORAGE	fb	Apple Mac. HFS+ filesystem
ID_SARC_DMG_STORAGE	fc	Apple Mac. DMG filesystem image
ID_SAVESET_STORAGE	110	VMS 'Saveset' archive
ID_VARIABLE	112	VMS variable-length text file

5 Data types

SOPHOS_TYPE_INVALID	Invalid type
SOPHOS_TYPE_U08	Unsigned byte (8 bits)
SOPHOS_TYPE_U16	Unsigned short (16 bits)
SOPHOS_TYPE_U32	Unsigned long (32 bits) - normally used for status settings
SOPHOS_TYPE_S08	Signed byte (8 bits)
SOPHOS_TYPE_S16	Signed short (16 bits)
SOPHOS_TYPE_S32	Signed long (32 bits)
SOPHOS_TYPE_BOOLEAN	Not used
SOPHOS_TYPE_BYTESTREAM	Not used
SOPHOS_TYPE_OPTION_GROUP	Group option (32 bits)
SOPHOS_TYPE_STRING	String

6 Technical support

For technical support, visit *http://www.sophos.com/support*.

If you contact technical support, provide as much information as possible, including the following:

- Sophos software version number(s)
- Operating system(s) and patch level(s)
- The exact text of any error messages

7 Copyright

Copyright © 2008 Sophos Group. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise unless you are either a valid licensee where the documentation can be reproduced in accordance with the licence terms or you otherwise have the prior permission in writing of the copyright owner.

Sophos and Sophos Anti-Virus are registered trademarks of Sophos Plc and Sophos Group. All other product and company names mentioned are trademarks or registered trademarks of their respective owners.